SalaMAnder: Shapley-based Mathematical Expression Attribution and Metric for Chain-of-Thought Reasoning

Yue Xin[1,2]*, **Chen Shen**[2]†‡, **Shaotian Yan**[2], **Xiaosong Yuan**[2],
Yaoming Wang[1], **Xiaofeng Zhang**[1,2]*, **Chenxi Huang**[2], **Jieping Ye**[2]

[1]Shanghai Jiao Tong University, [2]Alibaba Cloud Computing

Abstract

Chain-of-Thought (CoT) prompting enhances the math reasoning capability of large language models (LLMs) to a large margin. However, the mechanism underlying such improvements remains unexplored. In this paper, we present **SalaMAnder** (**S**h**a**p**l**ey-b**a**sed **M**athematical Expression **A**ttribution a**nd** M**e**t**r**ic), a theoretically grounded methodology as well as a mathematically rigorous evaluation metric for quantifying component-level contributions in few-shot CoT reasoning. Concretely, we leverage the Shapley value for mathematical expression attribution and develop an efficient stratified sampling algorithm that significantly reduces the computational complexity. Besides, we develop the **CoSP** (**C**ardinality **o**f **S**hapley **P**ositives) metric through covariance analysis. Comprehensive validation across popular LLM models and diverse mathematical benchmarks demonstrates that the CoSP metric within our SalaMAnder framework exhibits a robust monotonic correlation with model performance, not only providing theoretical explanations for the empirical success of existing few-shot CoT but also establishing mathematically rigorous principles for prompt construction optimization. Furthermore, we verify the reliability of the explanation, based on which we unify the insights of previous work.

1 Introduction

Chain-of-Thought (CoT) reasoning has elicited powerful mathematical ability within large language models (LLMs) reasoning tasks, ranging from arithmetic problem solving to theorem proving. Despite the substantial improvements, the mechanism of how reasoning steps lead to correct answers remains underexplored, both heuristic speculation(Wang et al., 2023; Chen et al., 2024; Wang et al., 2022; Li et al., 2024; Jin et al., 2024; Pfau et al., 2024) and labor-intensive verification(Serrano and Smith, 2019; Bastings and Filippova, 2020; Madsen et al., 2022; Siddiqui et al., 2024) lack theoretical investigation.

Prior heuristic-driven approaches analyze the role of different components by defining customized input formats. For instance, Chen et al. (2024) and Jin et al. (2024) introduce tailored reasoning steps during inference and investigate the impact of step order and length, respectively. While labor-intensive approaches attempt to explain CoT actions through ad hoc trial-and-error adjustments and case-specific manual inspections (Serrano and Smith, 2019; Bastings and Filippova, 2020; Madsen et al., 2022; Siddiqui et al., 2024). There is also a Shapley-value-based method (Horovicz and Goldshmidt, 2024) analyzing token-level attribution; nevertheless, the exponential computational complexity and indirect value function design hinder it from real-world applications.

In this paper, we propose a unified framework **SalaMAnder** (short for **S**h**a**p**l**ey-b**a**sed **M**athematical Expression **A**ttribution a**nd** M**e**t**r**ic), introducing two novel ideas for efficient and semantically coherent CoT analysis. First, we denote mathematical expressions as atomic units for Shapley-based attribution, addressing the semantic fragmentation inherent in traditional token-level analyses through component-level decomposition. Then, we develop a novel stratified sampling algorithm, namely **SalaMA (Shapley-based Mathematical Expression Attribution)** that achieves exponential complexity reduction by decomposing Shapley calculations according to component order, reducing time complexity from $O(2^{n+1})$ to $O(2mn^2)$ while maintaining rigorous theoretical guarantees, where n refers to the number of components and m indicates the number of samples. To supplement SalaMA, we also develop the **CoSP** (**C**ardinality **o**f **S**hapley **P**ositives) metric based on

[*] Work done during an internship at Alibaba Cloud Computing.

[†] Corresponding: Chen Shen<jason.sc@alibaba-inc.com>

[‡] Project Lead.

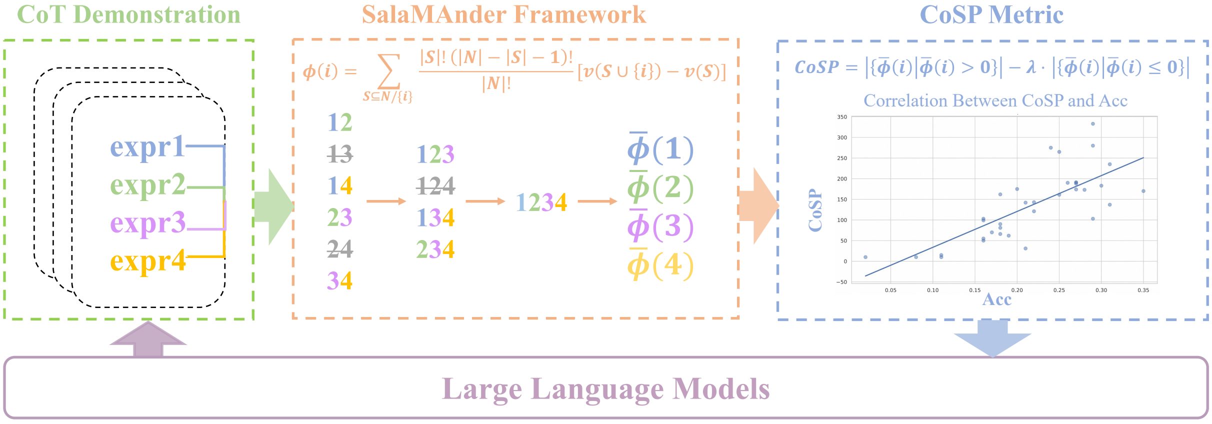


Figure 1: Workflow of the SalaMAnder Framework and CoSP Metric in CoT for LLMs. Initially, the framework proposes an efficient Shapley value algorithm to attribute the contributions of various mathematical expressions. These computed Shapley values are then utilized to derive the CoSP metric. Both theoretical derivations and extensive experiments across multiple models and datasets validate that CoSP exhibits a robust positive correlation with model inference accuracy. This correlation provides a comprehensive explanation of the underlying mechanisms driving CoT behavior in LLMs.

the efficient and semantical Shapley estimation.

The proposed CoSP metric within our SalaMAnder framework formally establishes the monotonic relationship with model performance. Theoretically, we provide a rigorous mathematical analysis of this monotonic relation. Experimentally, we apply SalaMAnder to few-shot learning scenarios, utilizing popular LLMs (LLaMA-2-13B-chat (Touvron et al., 2023), LLaMA-3-8B-Instruct (Grattafiori et al., 2024), and Qwen2.5-7B-Instruct (Team, 2024)) tested on various mathematical benchmarks (GSM8K (Ouyang et al., 2022), MathQA (Amini et al., 2019), AQUA (Ling et al., 2017), MultiArith (Wang et al., 2018), and SVAMP (Patel et al., 2021)) to compute the Pearson correlation coefficient. Then we further evaluate the reliability of the explanation results. Last, we present novel insights that not only reinforce the effectiveness of our methods but also integrate and unify previous research.

The contributions of this paper can be summarized as follows:

- We propose a unified framework SalaMAnder to establish mathematical expressions as atomic units for Shapley-based attribution, and we develop a novel stratified sampling algorithm SalaMA that achieves exponential complexity reduction while maintaining rigorous theoretical guarantees.
- We present the CoSP metric within our SalaMAnder framework, which formally establishes the monotonic relationship with model performance through rigorous covariance analysis, providing mathematical guarantees for the predictive validity.

2 Related Work

CoT Methodologies CoT prompting, introduced by Wei et al. (2022), explicitly guides LLMs to generate intermediate reasoning steps, significantly improving performance on mathematical and symbolic tasks. Subsequent work expanded this paradigm through path optimization (e.g., Least-to-Most prompting decomposes problems into subquestions (Zhou et al., 2022); Progressive-Hint iteratively refines solutions (Zheng et al., 2023)), automation (e.g., Automatic CoT generates demonstrations via LLMs (Zhang et al., 2022); Symbolic CoT Distillation transfers CoT ability to smaller models (Li et al., 2023)), and hybrid approaches (e.g., CoF-CoT combines coarse-to-fine prompting for multi-domain tasks (Nguyen et al., 2023); Deductive Verification adds formal consistency checks (Ling et al., 2023)). Despite these advances, most methods rely on heuristic designs without theoretical guarantees, and their efficacy varies significantly across domains—mathematical tasks benefit more from structured CoT than open-ended reasoning.

Mechanistic Studies of CoT Reasoning The existing literature on CoT mechanisms unfolds through complementary empirical and theoretical lenses. Empirical studies (Wang et al., 2022; Li et al., 2024; Jin et al., 2024; Wang et al., 2023; Pfau et al., 2024; Chen et al., 2024) have explored various strategies to enhance the robustness, safety, and structural integrity of CoT reasoning. For instance, self-consistency mechanisms (Wang et al., 2022) improve the reliability of reasoning outputs by aggregating multiple reasoning paths, while efforts to mitigate toxicity (Li et al., 2024) ensure safer commonsense reasoning. Additionally, research on step length (Jin et al., 2024), step relevance and logical order (Wang et al., 2023), hidden state dynamics (Pfau et al., 2024), and premise sequence order (Chen et al., 2024) underscores the importance of prompt design and structural factors in optimizing CoT performance.

Another set of literature attempts to explain CoT through ad hoc trial-and-error adjustments (Serrano and Smith, 2019; Bastings and Filippova, 2020; Madsen et al., 2022; Siddiqui et al., 2024). For instance, (Bastings and Filippova, 2020) and (Siddiqui et al., 2024) utilize attention maps and saliency score to analyze CoT, respectively. There is also a Shapley-value-based method (Horovicz and Goldshmidt, 2024) analyzing token-level attribution, nevertheless, the exponential computational complexity and indirect value function design hinder it from real-world applications.

3 Method

In this section, we introduce the **SalaMAnder** framework, designed to explain the mathematical reasoning mechanisms of CoT in LLMs using Shapley values. We introduce our method in three sections: an introduction to Shapley values, the **SalaMAnder** sparse computation of these values, and the **CoSP** metric for evaluating CoT reasoning contributions.

3.1 Preliminary: Shapley Values (Fair Attribution of CoT Constituents)

Shapley values, originating from cooperative game theory, offer a principled method for fairly distributing the total gains of a coalition among its individual players based on their contributions (Shapley, 1953).

Formally, consider a set of players $N = \{1, 2, \ldots, n\}$ and a reward function $v : 2^N \to \mathbb{R}$ that assigns a real-valued payoff to every possible coalition of players. The Shapley value $\phi_i(v)$ for player i is defined as:

$$\phi_v(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!} \left[v(S \cup \{i\}) - v(S) \right]$$

where S is any subset of N that does not include player i, and $s = |S|, n = |N|$ respectively denotes the number of players in subset S and set N.

We can further derive from the above expression:

$$\begin{aligned} \phi(i) &= \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{\binom{n-1}{s}} \left[v(S \cup \{i\}) - v(S) \right] \\ &= \frac{1}{n} \sum_{r=0}^{n-1} \mathbb{E}_{s=r} \left[v(S \cup \{i\}) - v(S) \right] \\ &= \frac{1}{n} \phi_{r+1}(i) \end{aligned} \tag{1}$$

where $\phi_k(i) = \mathbb{E}_{s=r} \left[v(S \cup \{i\}) - v(S) \right]$ denotes the $(r+1)$th order shapley value of component i.

Researchers have proven that the Shapley value is a unique unbiased method to fairly allocate overall reward to each player with four properties: linearity, dummy, symmetry, and efficiency (Weber, 1988). For simplicity, we use $\phi(i)$ by ignoring the superscript of $\phi_v(i)$ in the following manuscript without causing ambiguity.

In our framework, each component of the CoT, such as individual mathematical expressions or a single word, is treated as a player in the cooperative game. The reward function $v(S)$ corresponds to a performance metric of the LLM (e.g., correctness, or inference logits) when only the components in subset S are included in the CoT. Consequently, the Shapley value $\phi(i)$ quantifies the average marginal contribution of each component to the overall reasoning performance across all possible subsets of components.

The feasibility of our method is guaranteed by the non-essential requirement for independence among components in the mathematical definition of the Shapley value and the value function, although low feature independence truly has some problems. But in our specific context, expressions tend to carry distinct semantic roles, with rare high-level redundancy between them. Consequently, the computed Shapley values can effectively reflect the true contribution of each component in the reasoning process.

3.2 SalaMA: Efficient Sparse Shapley Computation for CoT Components

Although calculating exact Shapley values for each component presents significant computational challenges, the exponential growth in the number of possible subsets with respect to the number of components renders exact computation infeasible for practical applications. To address the limitation, we propose SalaMA (Shapley-based Mathematical Expression Attribution) mechanism, an efficient algorithm designed to approximate Shapley values with high accuracy while substantially reducing computational overhead.

The Players We define each player in the game, i.e. each component in the demonstration as a mathematical expression rather than individual words or tokens. This decision is motivated by the observation that single words or tokens can vary in meaning across different contexts, making their attribution inconsistent and less meaningful. Mathematical expressions, in contrast, maintain their semantic integrity across diverse reasoning scenarios, providing a more stable and universally applicable unit for analysis. Additionally, aggregating tokens into coherent mathematical expressions significantly reduces the number of components, thereby mitigating the computational complexity associated with Shapley value calculations. This aggregation not only enhances computational efficiency but also ensures that the attribution analysis remains interpretable and relevant to the model's problem-solving mechanisms.

The Reward Function We adopt a reward function that combines the model's prediction confidence logits with the correctness of the prediction, formulated as

$$\begin{cases} v(S) = \left(\frac{1}{L} \sum_{\ell=1}^{L} \log p_\theta(y_\ell | S) \right) \cdot \mathbb{I}(y_{\text{pred}}(S) = y^*) \\ y_{\text{pred}}(S) = \bigoplus_{\ell=1}^{L} y_\ell(S) \end{cases} \tag{2}$$

where $\frac{1}{L} \sum_{\ell=1}^{L} \log p_\theta(y_\ell|S)$ represents the average confidence score of the model's prediction by averaging the logits associated with the result tokens generated when including component subset S, $\mathbb{I}(\cdot)$ is a binary indicator, and $\bigoplus$ indicates the string concatenation operation.

This formulation ensures that the value function directly reflects the impact of each component on the model's performance, addressing the limitations of alternative metrics such as attention, saliency scores or binary correctness. Attention or saliency scores do not provide a direct attribution to the final outcome and can be complex to interpret (Serrano and Smith, 2019; Bastings and Filippova, 2020; Madsen et al., 2022; Siddiqui et al., 2024), while a binary correctness metric lacks the sensitivity needed to capture nuanced contributions. By integrating confidence logits with correctness, the reward function balances sensitivity and direct attribution, facilitating a more accurate and interpretable estimation of each component's contribution.

Efficient Shapley Computation Algorithm The proposed algorithm systematically approximates the Shapley values for CoT components through a structured algorithmic workflow. In exact Shapley value computation, for each component i, it is necessary to evaluate $v(S \cup \{i\}) - v(S)$ across all subsets $S \subseteq N\{i\}$, leading to a computational complexity of $O(2^{n+1})$, where n is the number of components. This exponential complexity becomes prohibitively expensive as the number of components increases. To mitigate this, SalaMA reduces the number of necessary inferences by employing a stratified sampling approach based on the order of Shapley values.

Specifically, the SalaMA mechanism decomposes the Shapley value calculation by order. For an r-th order Shapley value ϕ_r, SalaMA randomly samples $r-1$ other mathematical expressions from the set $N/\{i\}$. The number of such samples is denoted by sp, with a maximum limit of m, indicating $sp = \min(m, \binom{n-1}{r-1})$. In the original demonstration, aside from the mathematical expressions, other components (referred to as the "whiteboard") are always present and remain constant across different subsets.

During inference, for each sampled subset S of size $r-1$, SalaMA constructs two distinct demonstrations: one containing $S \cup \{i\}$, and another containing S, all combined with the whiteboard. These demonstrations are then fed into the model to obtain the corresponding reward functions $v(S \cup \{i\})$ and $v(S)$, respectively. By iterating over multiple orders and different samples within each order, SalaMA aggregates the marginal contributions across various subset configurations. The approxi-

Algorithm 1: SalaMA: Sparse Shapley Value Computation

```
Function SalaMA($N, v, n, m$):
    Initialize $\phi[i] \leftarrow 0\ (\forall i \in N),\ \mathcal{H} \leftarrow \emptyset$;
    foreach $i \in N$ do
        for $r = 1$ to $n$ do
            $sp \leftarrow \min(m, \binom{n-1}{r-1})$;
            for $s = 1$ to $sp$ do
                $S \leftarrow$ Sample$(r-1, N \setminus i)$;
                $v_S \leftarrow$ MemEval$(S, \mathcal{H})$;
                $v_{S \cup i} \leftarrow$ MemEval$(S \cup i, \mathcal{H})$;
                $\phi[i]$ += $(v_{S \cup i} - v_S)/(sp \cdot n)$;

            end
        end
    end
    return $\phi$;
Procedure MemEval($S, \mathcal{H}$):
    if $S \notin \mathcal{H}$ then
        $\mathcal{H}[S] \leftarrow v(S)$;
    end
    return $\mathcal{H}[S]$;
```

mated Shapley value can be derived from Eq. (1):

$$
\begin{aligned}
\phi(i) &= \frac{1}{n} \sum_{r=0}^{n-1} \mathbb{E}_{s=r}[v(S \cup \{i\} - v(S))] \\
&= \frac{1}{n} \sum_{r=0}^{n-1} \frac{1}{m} \sum_{t=1}^{m} [v(S_t^r \cup \{i\}) - v(S_t^r)] \quad (3)
\end{aligned}
$$

To further enhance computational efficiency, SalaMA maintains a hash table $\mathcal{H}$ to store and retrieve the results of previously computed subsets S. This caching mechanism avoids redundant inferences by storing $v(S)$ for each evaluated subset S. Consequently, the computational complexity of SalaMA is reduced to $O(2 \cdot sp \cdot n^2) \leqslant O(2mn^2)$, which is significantly lower than the exact Shapley value computation's $O(2^{n+1})$. The whole workflow is shown in Algorithm. 1. We also conduct experiments on the computation complexity and error magnitude of Shapley value in Appendix C, indicating that it is entirely feasible to achieve a trade-off between computational complexity and estimation accuracy with appropriate hyperparameter selection.

3.3 CoSP: Performance-Aligned Causal Explanation Rationale

We introduce CoSP (**C**ardinality **o**f **S**hapley **P**ositives), a metric defined as the number of expressions within a demonstration that exhibit positive average Shapley values minus a weighted non-positive average Shapley values across multiple experiments.

Formally, for a demonstration comprising a set of n expressions N, CoSP is defined as:

$$
\begin{aligned}
CoSP &= |\{\bar{\phi}(i) | \bar{\phi}(i) > 0\}| - \lambda \cdot |\{\bar{\phi}(i) | \bar{\phi}(i) \leqslant 0\}| \\
&= \sum_{i=1}^{n} \mathbb{I}(\bar{\phi}(i) > 0) - \lambda \cdot \mathbb{I}(\bar{\phi}(i) \leqslant 0) \\
&= (1+\lambda) \sum_{i=1}^{n} \mathbb{I}(\bar{\phi}(i) > 0) - \lambda n
\end{aligned}
$$

where $\bar{\phi}(i)$ is the average Shapley value of the i-th expression, computed over m different problem instances tested using the same demonstration, formulated as $\bar{\phi}(i) = \frac{1}{m} \sum_{k=1}^{m} \phi^{(k)}(i)$, $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the condition inside is true and 0 otherwise, and $\lambda > 0$ is the penalty severity for the number of expressions with negative Shapley values. And we assume that during the m CoT reasoning precesses, for each expression i, there is $\phi^{(k)}(i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

A positive average Shapley value ($\bar{\phi}(i) > 0$) indicates that the corresponding mathematical expression contributes positively to the model's reasoning performance; conversely, a non-positive one leads to negative contribution or no contribution. Therefore, CoSP comprehensively quantifies the number of expressions that actively enhance or degrade the model's efficacy in solving problems. A higher CoSP suggests that a greater subset of expressions within the CoT is beneficial while a smaller subset harmful, correlating with improved model performance. Specifically, we define CoSP-0 and CoSP-1, with λ equals to 0 and 1, respectively.

To substantiate the relationship between CoSP and performance, we formalize the following two theorems under specific statistical assumptions.

Theorem 1 *Both CoSP-0 and CoSP-1 have posi-*

	LLaMA 2 (↑)				LLaMA 3 (↑)				Qwen 2.5 (↑)			
	CoSP-0	CoSP-1	SSV	NoE	CoSP-0	CoSP-1	SSV	NoE	CoSP-0	CoSP-1	SSV	NoE
1-shot												
GSM8K	**0.76**	0.65	0.32	0.76	0.70	0.18	-0.14	**0.71**	**0.64**	0.62	0.54	0.43
MathQA	0.44	**0.62**	0.63	-0.08	0.37	**0.28**	0.19	0.10	-0.16	**0.28**	0.11	-0.22
AQUA	0.40	**0.46**	0.44	-0.31	-0.21	**0.48**	0.39	-0.40	-0.63	**-0.03**	-0.03	-0.67
MultiArith	**0.60**	0.52	0.02	0.53	**0.74**	0.44	0.44	0.09	0.78	0.71	**0.80**	-0.04
SVAMP	**0.49**	0.28	**0.21**	0.14	0.17	**0.21**	0.08	-0.35	**0.56**	0.50	0.56	-0.32
2-shot												
GSM8K	**0.75**	0.35	0.14	0.75	**0.49**	0.26	0.24	0.45	**0.80**	0.48	0.51	0.13
MathQA	0.36	**0.46**	0.35	-0.11	-0.20	**0.01**	0.07	-0.05	-0.20	-0.14	**-0.03**	-0.06
AQUA	**0.56**	0.51	0.48	-0.47	**0.09**	-0.04	-0.22	-0.50	0.22	0.52	**0.55**	-0.19
MultiArith	**-0.04**	-0.07	-0.20	-0.31	**0.82**	0.39	0.58	-0.24	**0.44**	0.18	0.16	0.06
SVAMP	**0.23**	0.05	-0.13	-0.02	**0.47**	0.44	-0.19	-0.17	**0.69**	0.61	0.53	-0.02
4-shot												
GSM8K	**0.77**	0.61	0.12	0.52	0.26	**0.37**	-0.15	-0.20	**0.80**	0.58	0.52	0.31
MathQA	**0.29**	-0.26	-0.46	-0.01	**0.40**	0.28	-0.02	-0.67	**0.18**	-0.33	-0.52	0.14
AQUA	**0.80**	0.77	-0.10	-0.11	-0.08	**0.20**	0.02	-0.19	-0.31	-0.11	**-0.05**	-0.43
MultiArith	**0.54**	0.33	0.42	0.22	**0.80**	0.23	-0.001	-0.47	**0.67**	0.51	0.24	-0.44
SVAMP	**0.63**	0.31	0.22	0.61	**0.10**	0.07	0.36	-0.17	**0.22**	-0.03	-0.14	-0.13
Average	**0.51**	0.37	0.16	0.14	**0.33**	0.25	0.11	-0.14	**0.31**	0.29	0.25	-0.10

Table 1: The correlation coefficients between different metrics and model inference accuracy across multiple datasets and models of few-shot tasks. For each dataset and each model, the largest correlation is **bolded**, indicating the best interpretation method. Here we use 'LLaMA 2', 'LLaMA 3', and 'Qwen2.5' in short for LLaMA-2-13B-chat(Touvron et al., 2023), LLaMA-3-8B-Instruct(Grattafiori et al., 2024), and Qwen2.5-7B-Instruct(Team, 2024).

tive correlation with the model performance:

$$
\begin{aligned}
&Cov(CoSP, Perf) = (1+\lambda)(\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i)\\
&Cov(Perf, CoSP\text{-}0) = (\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i)\\
&Cov(Perf, CoSP\text{-}1) = 2(\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i)
\end{aligned} \tag{4}
$$

where the meaning of δ_+, δ_-, X_i will be explained in the proof.

Theorem 2 *CoSP-0 has a positive correlation with the number of expressions* n*, while CoSP-1 has a negative correlation with* n*:*

$$
\begin{aligned}
\mathbb{E}[CoSP_{n+1}] &= (1+\lambda)\sum_{i=1}^{n+1} p_i - (n+1)\lambda\\
&= \mathbb{E}[CoSP_n] + p_{n+1} - \lambda
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
&\mathbb{E}[CoSP\text{-}0_{n+1}] - \mathbb{E}[CoSP\text{-}0_n] = p_{n+1} > 0\\
&\mathbb{E}[CoSP\text{-}1_{n+1}] - \mathbb{E}[CoSP\text{-}1_n] = p_{n+1} - 1 < 0
\end{aligned} \tag{6}
$$

The proof of Theo. 1 and Theo. 2 is applied in Appendix. A.

The number of expressions n in the CoT is often indicative of the complexity or difficulty of the reasoning task. Generally, increased reasoning difficulty generally leads to better model performance (OpenAI, 2024), provided that the additional complexity is constructively leveraged. Our Theo. 2 aligns with this observation by showing that a higher number of expressions n results in a higher CoSP-0, which in turn, per Theo. 1, correlates with enhanced model performance. This consistency underscores the validity of CoSP as a metric that not only accounts for the quantity of reasoning steps but also their qualitative impact on model efficacy.

4 Experiments

This section presents a comprehensive evaluation of the proposed SalaMAnder framework, demonstrating its applicability across various settings. Appendix B describes the experimental settings, and Sec 4.1 utilizes SalaMAnder in few-shot learning scenarios to assess the validity of our explanation method and metric. In Sec 4.2, we further evaluate the reliability of explanation results. Sec 4.3

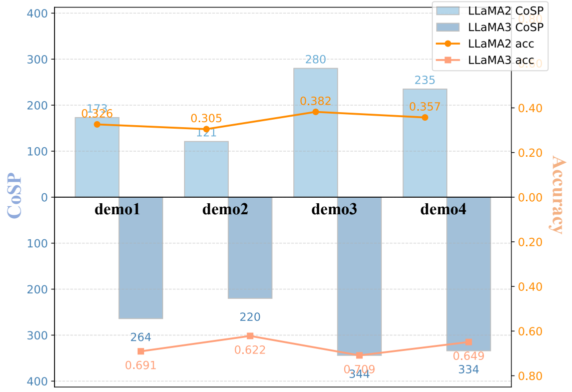


Figure 2: The CoSP-0 value and test accuracy of models. The strong consistency in their variation patterns further confirms the reliability of our explanation results.

presents novel insights that not only reinforce the effectiveness of our methods but also integrate and unify previous research, and Sec 4.4 provides qualitative analysis of the explanation.

Besides, we conduct experiments on the computation complexity and error magnitude of the calculation of Shapley value in Appendix C, indicating that it is entirely feasible to achieve a trade-off between computational complexity and estimation accuracy, thus guiding the selection of sample num. Appendix D illustrates ablation studies on the hyperparameters and Appendix E presents additional experimental results. And we show the cases used in Sec 4.3 in Appendix F, more cases in Appendix G, and the qualitative analysis cases in Appendix H.

4.1 Attribution Validity: CoSP Metric Verification in Few-Shot Learning

To evaluate the practical applicability of the proposed SalaMA method and the CoSP metric, we applied them to few-shot learning scenarios across multiple mathematical datasets and foundational language models to assess the correlation between CoSP and model performance (accuracy), thereby validating the effectiveness of our framework.

We meticulously constructed demonstrations to ensure a uniform distribution of mathematical expressions. Specifically, for one-shot learning tasks, we constructed demonstrations by selecting 35 question-answer (Q-A) pairs from the training sets of the GSM8K, MathQA, and AQUA datasets. Because the MultiArith and SVAMP datasets include answers composed solely of single mathematical expressions, we instead selected 35 Q-A pairs from the GSM8K dataset to serve as demonstrations. These one-shot demonstrations were evenly distributed, with five Q-A pairs each containing between one and seven mathematical expressions. For 2-shot demonstrations, the total number of expressions ranged from 2 to 10, resulting in 14 unique demonstrations by accounting for multiple combinations where applicable (e.g., a total of 6 expressions could be achieved by combinations 2+4 or 3+3). 4-shot demonstrations contained 4-16 total expressions, with one unique combination retained per expression count to minimize computation, producing 13 distinct demonstration sets. This methodology ensured that both one-shot and few-shot demonstrations maintained a balanced and uniform distribution of mathematical expressions, thereby isolating the effect of expression quantity on model performance.

We then utilize the proposed SalaMA method to few-shot learning to get various metrics: CoSP-0, CoSP-1, SSV (the sum of averaged shapley value, i.e. $\sum_{i=1}^{n} \bar{\phi}(i)$), NoE(number of expressions, i.e. n). The correlations of these metrics and model inference accuracy across diverse datasets and models in 1, 2, 4-shot scenarios are shown in Tab. 4, and Tab. 2 record the correlations averaged among different models.

Observed from Tab. 4, CoSP-0 is the best interpretation metric for all models, and the interpretation validity of CoSP-0/CoSP-1 is much better than the other metrics. According to Tab. 2, CoSP-0 serves as the best interpretation metric for GSM8K, MultiArith, and SVAMP, while CoSP-1 for AQUA. For MathQA, CoSP-0 serves as the best interpretation metric in 1 or 2-shot learning, while CoSP-1 the best in 4-shot learning.

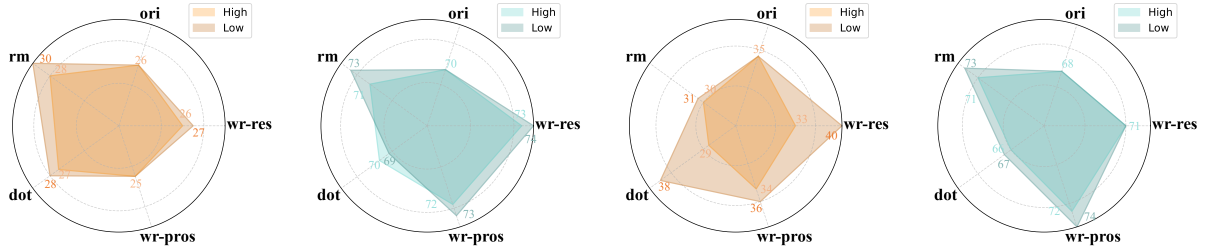
4.2 Explanation Reliability: Large-Scale Testing Assessment of CoSP Explanations

To further assess the reliability of our CoSP explanations, we conducted comprehensive validation experiments using the entire test set of the GSM8K dataset with both the LLaMA 2 and LLaMA 3 models. This focused approach ensures generality while maintaining computational feasibility. We selected four demonstrations for each model where the CoSP-0 scores for LLaMA 2 is 173, 121, 280, 235, while for LLaMA 3 is 264, 220, 344, 334.

The experimental outcomes consistently demonstrated a strong positive correlation between CoSP-0 scores and model accuracy for both LLaMA 2 and LLaMA 3 according to Fig. 2, with the experimental results of MathQA and AQUA shown in Fig. 4 in

	1-shot (↑)				2-shot (↑)				4-shot (↑)			
	CoSP-0	CoSP-1	SSV	NoE	CoSP-0	CoSP-1	SSV	NoE	CoSP-0	CoSP-1	SSV	NoE
GSM8K	**0.70**	0.48	0.24	0.63	**0.68**	0.36	0.33	0.44	**0.61**	0.52	0.16	0.21
MathQA	0.22	**0.39**	0.31	-0.07	-0.01	0.11	**0.13**	-0.07	**0.29**	-0.10	-0.33	-0.18
AQUA	-0.15	**0.30**	0.27	-0.46	0.29	**0.33**	0.27	-0.39	0.14	**0.29**	-0.04	-0.24
MultiArith	**0.71**	0.56	0.02	0.42	**0.41**	0.17	0.18	-0.16	**0.64**	0.36	0.22	-0.23
SVAMP	**0.41**	0.33	0.28	-0.18	**0.46**	0.37	0.07	-0.07	**0.32**	0.12	0.15	0.10

Table 2: The correlation coefficients averaged among various models in few-shot tasks. For each dataset, the largest correlation is **bolded**, indicating the best interpretation method.



(a) LLaMA2 demo1 (b) LLaMA3 demo1 (c) LLaMA2 demo2 (d) LLaMA3 demo2

Figure 3: Accuracy of demonstrations for low and high CoSP-0 expressions after four types of modifications in the test set across different models and demos: (a) LLaMA2-demo1, (b) LLaMA3-demo1, (c) LLaMA2-demo2, and (d) LLaMA3-demo2. The observed results indicate that the accuracy curve for low CoSP-0 expressions encompasses that for high CoSP-0 expressions in almost all scenarios, highlighting that alterations on low CoSP-0 expressions yield overall better performance outcomes compared to alterations on high CoSP-0 expressions.

Appendix E. Specifically, for LLaMA 2, the demonstration with a CoSP-0 score of 280 achieved the highest accuracy, followed by demonstrations with scores of 235, 173, and 121, in descending order of performance. Similarly, for LLaMA 3, the demonstration with a CoSP-0 score of 344 yielded the highest accuracy, followed by those with scores of 334, 264, and 220. This consistent pattern across both models indicates that demonstrations with higher CoSP-0 scores significantly enhance the reasoning capabilities of the models, while those with lower scores contribute less effectively.

To be mentioned, the strong consistency in CoSP-0 and model accuracy not only confirms the reliability of the explanation results provided by SalaMAnder, but also reveals a potential application in the systematic selection of few-shot demonstrations, rather than random sampling.

4.3 Analytical Extensibility: Discovery of Novel Insights in CoT

Building upon our previous findings that high CoSP expressions contribute maximally, while low ones contribute minimally to model reasoning, we sought to uncover novel insights into the dynamics of CoT reasoning processes. Specifically, we applied four distinct altering to the expression with the highest and lowest CoSP-0 to assess their impact on model performance. 1) Removed the expression. 2) Replaced the expressions with non-informative placeholders, i.e., '...'. 3) Introduced calculation errors, for example, converting from '2 + 3 = 5' to '2 + 3 = 6'. 4) Introduced process errors, for example, converting from '2 + 3 = 5' to '4 + 7 = 11'. And we selected two demonstrations and conducted these experiments on GSM8K datasets, with both the LLaMA 2 and LLaMA 3 models. The original demonstration is presented in Appendix F, where different expressions of CoSP in different colors. Additional experiments on MathQA and AQUA are illustrated in Appendix E and more cases are shown in Appendix G for reference.

Figures 3 depict the effect of these alterations on the accuracy of the test set for low and high CoSP expressions across different demonstrations and models. It was consistently observed across almost all experiments that the performance curves for low CoSP expressions encapsulated those for high CoSP expressions.

The results suggest that modifications to low CoSP expressions lead to better performance outcomes compared to modifications to high CoSP expressions. This finding further corroborates our initial hypothesis: low CoSP expressions exert minimal influence on model reasoning, whereas high ones significantly contribute.

Additionally, our experimental findings reveal several intriguing phenomena. Notably, the removal of certain expressions, the substitution of expressions with non-informative filler tokens (such as '...'), and the introduction of errors in either the result or process of expressions do not necessarily lead to significant degradation in model performance. This outcome resonates with prior studies(Pfau et al., 2024; Wang et al., 2023).

4.4 Qualitative Analysis

We qualitatively illustrate why some demonstrations have higher CoSP values and are beneficial for model reasoning, while others are not. We analyze both the entire demonstration level and the individual expression level.

As for the CoSP of the entire demonstration, consider Example 1 and Example 2: the first contains more expressions and exhibits a richer logical structure, thus providing more informative signals for the model's reasoning. The second involves simpler computations (only division and comparison operations) and fails to convey a meaningful reasoning pattern, resulting in a smaller positive contribution. These examples can be found in Appendix H.

For the individual expression level, consider Example 3, where the expression $6/3 = 2$ has a low CoSP value, whereas $=> (x_3+x_4+x_5+x_6)/4 = 85$ has a high one. The former computes an irrelevant intermediate variable (in this case, one third of the strings), which does not contribute meaningfully to the final answer. In contrast, the latter is directly linked to the final solution and builds upon previous computations, making it highly relevant to the reasoning process, thus yielding a higher CoSP. Example 3 is shown below:

Example3

```
Question:
the average length of 6 strings is
80 cm. if the average length of one
third of the strings is 70 cm, what
is the average of the other strings ?
a ) 75. , b ) 85. , c ) 90. , d ) 94.
, e ) 100.
Answer:
edit : given
( x1 + x2 ... + x6 ) / 6 = 80
( x1 + x2 ... + x6 ) = 480 - -
> eq 1 . now given avg length
of one third strings is 70
. that means out 6 / 3 = 2
strings. let the avg length of
two strings be ( x1 + x2 ) / 2 = 70
. ( x1 + x2 ) = 140 . - - > eq
2 . now we are asked to find
the average of the remaining i
. e . ( x3 + x4 + x5 + x6 )
substitute eq 2 in eq 1 then we
get 140 + x3 + x4 + x5 + x6 = 480
= > x3 + x4 + x5 + x6 = 340 now
divide 340 by 4 we get 85 .
=> ( x3 + x4 + x5 + x6 ) / 4 = 85 =
avg length of remaining strings . the
correct option is b.
```

5 Conclusion

In this paper, we propose **SalaMAnder**, a novel framework for understanding and optimizing Chain-of-Thought (CoT) reasoning in large language models (LLMs). By introducing a theoretically grounded methodology based on Shapley value attribution and developing the **CoSP (Cardinality of Shapley Positives)** metric, we have established a mathematically rigorous approach to quantifying component-level contributions in CoT reasoning. Extensive validation across various LLM models and mathematical benchmarks demonstrates that the CoSP metric within our SalaMAnder framework strongly and monotonically correlates with model performance. This correlation not only theoretically explains the empirical success of the existing few-shot CoT but also provides rigorous guidelines for optimizing prompt construction. Furthermore, it can be utilized to discover novel insights resonating with prior studies.

Limitations

While SalaMAnder is theoretically a general approach, we are currently focusing on mathematical reasoning problems because they are highly representative of few-shot CoT reasoning. In the future we aim to expand the application of SalaMAnder to a broader array of tasks.

Due to computational resource constraints, our experiments are currently confined to LLMs with a parameter scale between 7 billion and 13 billion.

Acknowledgments

This work was supported by Alibaba Research Intern Program.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise Order Matters in Reasoning with Large Language Models. *arXiv e-prints*, arXiv:2402.08939.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, and Tobias Speckbacher. 2024. The Llama 3 Herd of Models. *arXiv e-prints*, arXiv:2407.21783.

Miriam Horovicz and Roni Goldshmidt. 2024. TokenSHAP: Interpreting large language models with Monte Carlo shapley value estimation. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 1–8, Miami, FL, USA. Association for Computational Linguistics.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The Impact of Reasoning Step Length on Large Language Models. *arXiv e-prints*, arXiv:2401.04925.

Jiachun Li, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024. Focus on Your Question! Interpreting and Mitigating Toxic CoT Problems in Commonsense Reasoning. *arXiv e-prints*, arXiv:2402.18344.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic Chain-of-Thought Distillation: Small Models Can Also “Think” Step-by-Step. *arXiv e-prints*, arXiv:2306.14050.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive Verification of Chain-of-Thought Reasoning. *arXiv e-prints*, arXiv:2306.03872.

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1731–1751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hoang H. Nguyen, Ye Liu, Chenwei Zhang, Tao Zhang, and Philip S. Yu. 2023. CoF-CoT: Enhancing Large Language Models with Coarse-to-Fine Chain-of-Thought Prompting for Multi-domain NLU Tasks. *arXiv e-prints*, arXiv:2310.14623.

OpenAI. 2024. Openai o1 system card. [Online]. `https://cdn.openai.com/o1-system-card-20241205.pdf`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Jacob Pfau, William Merrill, and Samuel R. Bowman. 2024. Let's Think Dot by Dot: Hidden Computation in Transformer Language Models. *arXiv e-prints*, arXiv:2404.15758.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2:307–317.

Shoaib Ahmed Siddiqui, Radhika Gaonkar, Boris Köpf, David Krueger, Andrew Paverd, Ahmed Salem, Shruti Tople, Lukas Wutschitz, Menglin Xia, and Santiago Zanella Béguelin. 2024. Permissive information-flow analysis for large language models. *ArXiv*, abs/2410.03055.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints*, arXiv:2307.09288.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018. Mathdqn: solving arithmetic word problems via deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv e-prints*, arXiv:2203.11171.

Robert James Weber. 1988. *Probabilistic values for games*, page 101–120. Cambridge University Press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models. *arXiv e-prints*, arXiv:2210.03493.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-Hint Prompting Improves Reasoning in Large Language Models. *arXiv e-prints*, arXiv:2304.09797.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. *arXiv e-prints*, arXiv:2205.10625.

A The proof of Theorems

We have three assumptions necessary for the proof:

1. The positive contribution of any expression has a significant lower bound:

$$\begin{aligned}&\exists\ \delta_+ > 0,\ s.t.\\ &\mu_i > \delta_+ \cdot \mathbb{I}(\mu_i > 0)\end{aligned}$$

2. The non-positive contribution of any expression has a lower bound:

$$\begin{aligned}&\exists\ \delta_- < 0,\ s.t.\\ &\mu_i > \delta_- \mathbb{I}(\mu_i \leqslant 0) = \delta_- \cdot (1 - \mathbb{I}(\mu_i > 0))\end{aligned}$$

3. The contributions of different expressions are mutually independent when applied to different problems:

$$\begin{aligned}&\mathrm{Cov}(\phi^{(k)}(i), \phi^{(l)}(j)) = 0\\ &(\forall\ i \neq j, 1 \leqslant k, l \leqslant m, k \neq l)\end{aligned}$$

Here is the proof of Theo. 1:

Proof 1 *As illustrated in Sec. 3.3:*

$$\begin{aligned}&\phi^{(k)}(i) \sim \mathcal{N}(\mu_i, \sigma_i^2)\\ &\bar{\phi}(i) = \frac{1}{m}\sum_{k=1}^{m} \phi^{(k)}(i) \overset{m\to\infty}{\longrightarrow} \mu_i\end{aligned}$$

To simplify the expression, we define a positive contribution indicator $X_i = \mathbb{I}(\bar{\phi}(i) > 0)$. Thus:

$$\begin{aligned}CoSP &= \sum_{i=1}^{n} X_i - \lambda \sum_{i=1}^{n}(1 - X_i)\\ &= (1+\lambda)\sum_{i=1}^{n} X_i - n\lambda \end{aligned} \tag{7}$$

And we define the model performance $Perf$ by summing the expected shapley value of all expressions:

$$Perf = \sum_{i=1}^{n} \mathbb{E}[\phi(i)] = \sum_{i=1}^{n} \mu_i \tag{8}$$

Thus we can further derive the expression of $Perf$:

$$\begin{aligned}Perf &= \sum_{i\in S_+} \mu_i + \sum_{i\notin S_+} \mu_i > \sum_{i\in S_+} \delta_+ + \sum_{i\notin S_+} \delta_-\\ &= \sum_{i=1}^{n} \delta_+ \mathbb{I}(\mu_i > 0) + \sum_{i=1}^{n} \delta_- \mathbb{I}(\mu_i \leqslant 0)\\ &= \sum_{i=1}^{n} \delta_+ \mathbb{I}(\mu_i > 0) + \sum_{i=1}^{n} \delta_-(1 - \mathbb{I}(\mu_i > 0))\\ &= n\delta_- + \sum_{i=1}^{n}(\delta_+ - \delta_-) \cdot \mathbb{I}(\mu_i > 0)\\ &= n\delta_- + (\delta_+ - \delta_-) \cdot \frac{CoSP + n\lambda}{1+\lambda} \end{aligned} \tag{9}$$

indicating a linear functional relationship between a lower bound of model performance and $CoSP$.

And the coveriance between μ_i and X_i is:

$$Cov(\mu_i, X_i) = \mathbb{E}[\mu_i X_i] - \mathbb{E}[\mu_i]\mathbb{E}[X_i]$$

where $\mu_i > \delta_+ X_i + \delta_-(1 - X_i)$ based on the first two assumptions.

We define a residual item $\epsilon_i > 0$, s.t. :

$$\mu_i = \delta_+ X_i + \delta_-(1 - X_i) + \epsilon_i$$

Then

$$\begin{aligned}\mathbb{E}[\mu_i X_i] &= \delta_+ \mathbb{E}[X_i^2] + \delta_- \mathbb{E}[(1 - X_i)X_i] + \mathbb{E}[\epsilon_i X_i]\\ &= \delta_+ + \mathbb{E}[\epsilon_i X_i]\end{aligned}$$

The second equation is because $X_i(1 - X_i) = 0$.

And

$$\mathbb{E}[\mu_i] = \delta_+ \mathbb{E}[X_i] + \delta_- \mathbb{E}[1 - X_i] + \mathbb{E}[\epsilon_i]$$

Thus

$$\begin{aligned}Cov(\mu_i, X_i) =&\delta_+ \mathbb{E}[X_i^2] + \mathbb{E}[\epsilon_i X_i] - \delta_+ \mathbb{E}^2[X_i] -\\ &\delta_- \mathbb{E}[X_i]\mathbb{E}[1 - X_i] + \mathbb{E}[X_i]\mathbb{E}[\epsilon_i]\end{aligned}$$

Since $\mathbb{E}[X_i] = \mathbb{E}[X_i^2]$, and $\mathbb{E}[1 - X_i] = 1 - \mathbb{E}[X_i]$, then

$$\begin{aligned}\mathbb{E}[X_i]\mathbb{E}[1 - X_i] &= \mathbb{E}[X_i](1 - \mathbb{E}[X_i])\\ &= \mathbb{E}[X_i] - \mathbb{E}^2[X_i]\\ &= \mathbb{E}[X_i^2] - \mathbb{E}^2[X_i]\\ &= Var(X_i)\end{aligned}$$

Then

$$Cov(\mu_i, X_i) = (\delta_+ - \delta_-)Var(X_i) + Cov(\epsilon_i, X_i) \tag{10}$$

Based on the third assumption, we have:

$$\begin{aligned}
&\mathit{Cov}(Perf, CoSP) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathit{Cov}(\mu_i, (1+\lambda)X_j - \lambda) \\
&= \sum_{i=1}^{n} \mathit{Cov}(\mu_i, (1+\lambda)X_i - \lambda) \\
&= (1+\lambda)\sum_{i=1}^{n} \mathit{Cov}(\mu_i, X_i) \\
&= (1+\lambda)\left[(\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i) + \sum_{i=1}^{n} \mathit{Cov}(\epsilon_i, X_i)\right]
\end{aligned}$$

And since the residual ϵ_i has little relevance with X_i, the sum of the covariance tends to 0. Thus

$$\begin{aligned}
\mathit{Cov}(Perf, CoSP) &= (1+\lambda)(\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i) \\
&> 0 \qquad (11)
\end{aligned}$$

Specifically, we define CoSP-0 and CoSP-1, with λ equals to 0 and 1, respectively. Then

$$\mathit{Cov}(Perf, CoSP\text{-}0) = (\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i) \qquad (12)$$

$$\mathit{Cov}(Perf, CoSP\text{-}1) = 2(\delta_+ - \delta_-)\sum_{i=1}^{n} \mathit{Var}(X_i) \qquad (13)$$

Thus $CoSP$ has a positive correlation with model performance.

□

Here is the proof of Theo. 2:

Proof 2 *Since $X_i = \mathbb{I}(\bar{\phi}(i) > 0)$, then X_i follows a Bernoulli distribution:*

$$p_i = P(X_i = 1) = \Phi(\frac{\mu_i}{\sigma_i}) \qquad (14)$$

where $\Phi(\cdot)$ is the standard normal distribution cumulative function.

Thus the expected value of $CoSP$ with n expressions is:

$$\begin{aligned}
\mathbb{E}[CoSP_n] &= (1+\lambda)\sum_{i=1}^{n} \Phi(\frac{\mu_i}{\sigma_i}) - n\lambda \\
&= (1+\lambda)\sum_{i=1}^{n} p_i - n\lambda \qquad (15)
\end{aligned}$$

Thus the expected value of $CoSP$ with $n+1$ expressions is:

$$\begin{aligned}
\mathbb{E}[CoSP_{n+1}] &= (1+\lambda)\sum_{i=1}^{n+1} p_i - (n+1)\lambda \\
&= \mathbb{E}[CoSP_n] + p_{n+1} - \lambda \qquad (16)
\end{aligned}$$

Therefore, CoSP-0 increases monotonically with the number of expressions n, while CoSP-1 decreases monotonically with n.

□

B Experimental Settings

To evaluate the effectiveness of the proposed SalaMA method and the CoSP metric, we conducted experiments using three foundational large language models and five representative mathematical datasets. The selected models, LLaMA-2-13B-chat (Touvron et al., 2023), LLaMA-3-8B-Instruct (Grattafiori et al., 2024), and Qwen2.5-7B-Instruct (Team, 2024) were drawn from various model families, each featuring distinct architectures and parameter sizes. This ensures that our analysis of CoSP and SalaMA is broadly applicable across different model paradigms.

For the datasets, we utilized GSM8K (Ouyang et al., 2022), MathQA (Amini et al., 2019), AQUA (Ling et al., 2017), MultiArith (Wang et al., 2018), and SVAMP (Patel et al., 2021). These datasets were selected for their representativeness in the mathematical question-answering domain, encompassing a range of difficulties where MathQA and AQUA are approximately equivalent and more challenging than GSM8K, which is in turn more difficult than MultiArith and SVAMP. Specifically, GSM8K consists of grade-school level math problems, MathQA includes complex multi-step reasoning questions, AQUA focuses on arithmetic and algebraic tasks, MultiArith provides multi-step arithmetic word problems, and SVAMP introduces adversarial variations to traditional arithmetic problems. This selection ensures comprehensive coverage of various aspects and complexities inherent in mathematical QA tasks.

C The Trade-off Between Computation Complexity and Error Magnitude

As illustrated before, The computational complexity of the Shapley value is $O(2^{n+1})$, while the complexity of our proposed SalaMa method is $O(2mn^2)$ where m denotes the number of samples, and n indicates the number of mathematical expressions. We evaluate the model inference cost and the relative error between the estimated and true Shapley values under different sampling settings. We randomly select a demonstration with $n = 8$ to illustrate the trade-off between efficiency and accuracy, where the maximum number of combinations at each order is $\binom{7}{3} = 35$ according to Eq. (1).

m	5	15	25	35
error(%)	62	45	12	0

Table 3: The computation complexity and relative error of Shapley value.

As shown in Tab. 3, it is entirely feasible to achieve a trade-off between computational complexity and estimation accuracy by selecting appropriate hyperparameters. For example, setting the sample number to 25 allows us to significantly reduce the computational cost while maintaining high precision in Shapley value estimation.

D Ablation Study on Hyperparameters

In this section, we conduct ablation studies on the hyperparameter λ, which indicates the penalty to the mathematical expressions of negative contribution. $\lambda = 0$ shows no penalty and only encourages positive contributions, while $\lambda = 1$ demonstrates that equal attention is given to both positive and negative contributions. Thus a feasible value of λ is in $[0, 1]$. Here we conduct this ablation study on LLaMA2-13B on various mathematical datasets, with the same experimental setup as in Sec 4.1.

	LLaMA 2 (↑)			
	CoSP-0	CoSP-0.5	CoSP-0.8	CoSP-1
GSM8K	0.76	**0.77**	0.75	0.65
MathQA	0.44	0.58	0.59	**0.62**
AQUA	0.40	0.49	**0.53**	0.46
MultiArith	**0.60**	0.60	0.54	0.52
SVAMP	**0.49**	0.34	0.27	0.28

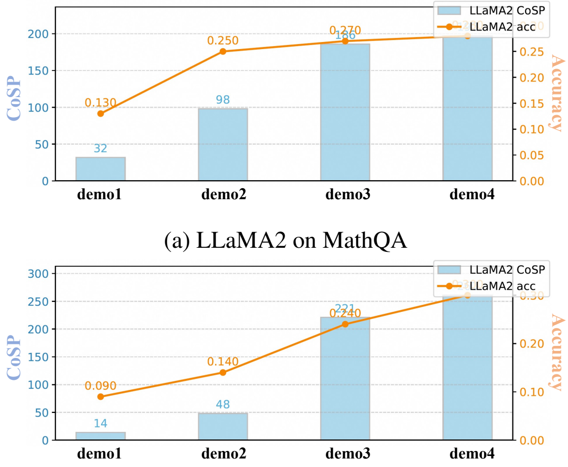
Table 4: The correlation coefficients between different metrics and model inference accuracy across multiple datasets on LLaMA2-13B-chat on 1-shot task.

As can be observed, when λ is set within the range $(0, 1)$, the correlation between CoSP-λ and model accuracy lies between that of CoSP-0 and CoSP-1 on almost all datasets, and it changes almost monotonically with λ. This suggests a smooth transition in the positive and negative contributions as the penalty weight is adjusted, further supporting the robustness of our approach under different weighting schemes.

E Additional Experiments

Additional experiments of Sec 4.2 and Sec 4.3 are shown here.

Fig 4 exhibits the results of LLaMA2 conducted on MathQA and AQUA, with the same setups as Sec 4.2. Fig. 4 illustrates strong consistency in the model performance and CoSP-0.

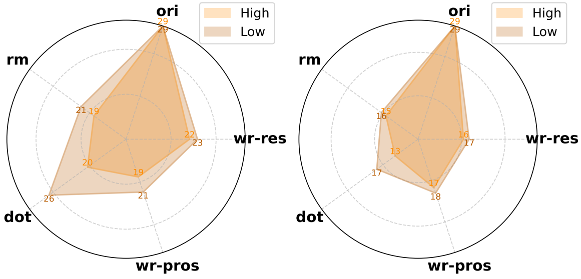


(a) LLaMA2 on MathQA

(b) LLaMA2 on AQUA

Figure 4: The CoSP-0 value and test accuracy of models. (a) LLaMA2 on MathQA, (b) LLaMA on AQUA.

Fig 5 shows the results of LLaMA2 conducted on MathQA and AQUA, with the same setups as Sec 4.3.



(a) LLaMA2 on MathQA

(b) LLaMA2 on AQUA

Figure 5: Accuracy of demonstrations for low and high CoSP-0 expressions after four types of modifications in the test set across different models and demos. (a) LLaMA2 on MathQA, (b) LLaMA on AQUA.

Fig. 5 indicates that the accuracy curve for low CoSP-0 expressions encompasses that for high CoSP-0 expressions in almost all scenarios, highlighting that alterations on low CoSP-0 expressions yield overall better performance outcomes compared to alterations on high CoSP-0 expressions.

F Selected Demonstrations

This section presents the selected demonstrations in Sec 4.3. Expressions with a light blue background have the lowest CoSP, those with an orange background have the highest CoSP, and the remaining expressions are shown with a light green background.

demo1

Question:
Sharon wants to get kitchen supplies. She admired Angela's kitchen supplies which consist of: 20 pots, 6 more than three times as many plates as the pots, and half as many cutlery as the plates. Sharon wants to buy: half as many pots as Angela, 20 less than three times as many plates as Angela, and twice as much cutlery as Angela. What is the total number of kitchen supplies Sharon wants to buy?
Answer:
Angela has 6+3*20=«6+3*20=66»66 plates. Angela has 1/2*66=«1/2*66=33»33 cutlery. Sharon wants to buy 1/2*20=«1/2*20=10»10 pots. Sharon wants to buy 3*66-20=«3*66-20=178»178 plates. Sharon wants to buy 2*33=«2*33=66»66 cutlery. Sharon wants to buy a total of 10+178+66=«10+178+66=254»254 kitchen supplies.

demo2

Question:
Brittany, Alex, and Jamy all share 600 marbles divided between them in the ratio 3:5:7. If Brittany gives Alex half of her marbles, what's the total number of marbles that Alex has?
Answer:
The total ratio representing the number of marbles is 3+5 +7 = «3+5+7=15»15. From the ratio, the fraction representing the number of marbles that Brittany has is 3/15, which is equal to 3/15*600 = «3/15*600=120»120 marbles.Alex has 5/15*600 = «5/15*600=200»200 marbles.If Brittany gives half of her marbles to Alex, Alex receives 1/2*120 = 60 marbles.After receiving 60 marbles from Brittany, Alex has 200+60 = «200+60=260»260 marbles.

G More Cases

This section presents more demonstrations, with mathematical expressions with different CoSP shaded in different colors. The shading rule is the same as Appendix F, where the expressions with highest, medium, and lowest CoSP are shaded in orange, light green, and light blue.

Additionally, we demonstrate that there is minimal bias in CoSP and the positioning of expressions. Intuitively, expressions that are closer to the final answer tend to be more important, as they directly guide the model toward a specific output. In contrast, earlier expressions often represent intermediate or preliminary steps, which may have less influence on the final outcome. So the contribution of expressions may have bias with their positions. However, the CoSP of the expression shows low correlation with its position in the demonstration in most cases, indicating the feasibility of our framework.

demo3

Question:
Sasha added 48 cards into a box. Her sister, Karen, then took out 1/6 of the cards Sasha added. If there are now 83 cards in the box, how many cards were originally in the box?

```
Answer:
Karen took out 48/6 = «48/6=8»8 cards from the box.
Originally, the box had 83-40 = «83-40=43»43 cards.
```

demo4

```
Question:
Coleen loved sprinkles. At the beginning of the day, she had twelve cans of sprinkles. After applying sprinkles to her hair, her clothing and her pets, she had 3 less than half as many cans of sprinkles as she started out with. How many cans of sprinkles remained?
Answer:
Half of twelve cans of sprinkles is 12/2=«12/2=6»6 cans.
Three less than half as many cans of sprinkles is 6-3=«6-3=3»3 cans of sprinkles.
```

demo5

```
Question:
Ali is collecting bottle caps. He has 125 bottle caps. He has red ones and green ones. If he has 50 red caps, what percentage of caps are green?
Answer:
He has 75 green caps because 125 - 50 = «125-50=75»75
The proportion of caps that are green is .6 because 75 / 125 = «75/125=.6».6
The percentage that are green is 60 because .6 x 100% = «60=60»60%
```

demo6

```
Question:
Nathan plays amateur baseball. He played for 3 hours for two weeks, every day. His friend Tobias played for 5 hours every day, but only for one week. How many hours did Nathan and Tobias play in total?
Answer:
Two weeks are 14 days, so Nathan played for 3 * 14 = «14*3=42»4 hours.
Tobias played for 7 days, so he played a total of 5 * 7 = «5*7=35»35 hours.
Nathan and Tobias played together for 42 + 35 = «42+35=77»77 hours.
```

demo7

```
Question:
While bird watching, Gabrielle saw 5 robins, 4 cardinals, and 3 blue jays. Chase saw 2 robins, 3 blue jays, and 5 cardinals. How many more birds, in percentage, did Gabrielle saw than Chase?
Answer:
Gabrielle saw 5 + 4 + 3 = «5+4+3=12»12 birds.
Chase saw 2 + 3 + 5 = «2+3+5=10»10 birds.
So, Gabrielle saw 12 - 10 = «12-10=2»2 more birds than Chase.
Therefore, Gabrielle saw 2/10 x 100% = 20% more birds than Chase.
```

demo8

Question:

Two alien spacecraft on a sightseeing tour of Earth left New Orleans airport at 3:00 pm to travel the 448-mile distance to Dallas by air. Traveling nonstop, the first spacecraft landed in Dallas at 3:30 pm, while the second spacecraft landed in Dallas thirty minutes later. Assuming both spacecraft traveled at constant speed, what was the difference in speed, in miles per hour, between the two spacecraft?

Answer:

The first spacecraft flew for 30 minutes, or 30/60=1/2 hour.

The second spacecraft flew for 30+30=«30+30=60»60 minutes, or 1 hour.

Thus the first spacecraft traveled at a speed of 448 miles in 1/2 hour, or 448/(1/2)=896 miles per hour.

The second spacecraft traveled 448 miles in 1 hour, or 448/1=«448/1=448»448 miles per hour.

The difference in speed, in miles per hour, between the two spacecraft was 896-448=«896-448=448»448 miles per hour.

demo9

Question:

Julio has four bottles of orange soda and seven bottles of grape soda in his fridge. His friend Mateo has a bottle of orange soda and 3 bottles of grape soda in his fridge. If the amount of beverage in each bottle is 2 liters, how many more liters of soda does Julio have?

Answer:

Julio has 4 * 2 = «4*2=8»8 liters of orange soda

Julio also has 7 * 2 = «7*2=14»14 liters of grape soda.

Julio therefore has a total of 8 + 14 = «8+14=22»22 liters of soda

The amount of orange soda that Mateo has is 1 * 2 = «1*2=2»2 liters of orange soda

In addition, Mateo has 3 * 2 = «3*2=6»6 liters of grape soda.

In total, Mateo has 2 + 6 = «2+6=8»8 liters of soda.

This means that Julio has 22 - 8 = «22-8=14»14 liters more of soda

demo10

Question:

In a class of 30 students, the teacher polls the students on their favorite subject. 1/5 of the students like Math, and 1/3 like English. 1/7 of the remaining students like Science. The rest don’t have a favorite subject. How many students don’t have a favorite subject?

Answer:

30 x 1/5 = «30*1/5=6»6 students like Math.

30 x 1/3 = «30*1/3=10»10 students like English.

So, 6 + 10 = «6+10=16»16 students like either Math or English.

Thus, 30 - 16 = «30-16=14»14 students neither like Math nor English.

Since 1/7 of the remaining like Science, therefore 14 x 1/7 = «14*1/7=2»2 students like Science.

Hence, 14 - 2 = «14-2=12»12 students neither likes the 3 subjects.

H Qualitative Analysis Examples

This section presents Example 1 and Example 2 used in Sec 4.4.

Example1

Question:

a, b, k start from the same place and travel in the same direction at speeds of 30 km / hr, 40 km / hr, 60 km / hr respectively. b starts three hours after a. if b and k overtake a at the same instant, how many hours after a did k start? a) 3 , b) 4.5 , c) 6 , d) d) 5.5 , e) e) 5

Answer:

”the table you made doesn’t make sense to me. all three meet at the same point means the distance they cover is the same. we know their rates are 30, 40 and 60. say the time taken by b is t hrs. then a takes 3 + t hrs. and we need to find the time taken by k. distance covered by a = distance covered by b 30 * (3 + t) = 40 * t t = 9 hrs distance covered by b = distance covered by k 40 * t = 60 * time taken by k time taken by k = 40 * 9 / 60 = 6 hrs time taken by a = 3 + t = 3 + 9 = 12 hrs time taken by k = 6 hrs so k starts 12 - 6 = 6 hrs after a . (answer c)”

Example2

Question:

Question: of 70 players on a football team, 46 are throwers. the rest of the team is divided so one third are left - handed and the rest are right handed. assuming that all throwers are right handed, how many right - handed players are there total? a) 54 , b) 59 , c) 63 , d) 71 , e) 62

Answer:

”total = 70 thrower = 46 rest = 70 - 46 = 24 left handed = 24 / 3 = 8 right handed = 16 if all thrower are right handed then total right handed is 46 + 16 = 62 so e. 62 is the right answer”